Fractional Collections with Cardinality Bounds, and Mixed Linear Arithmetic with Stars

Ruzica Piskac and Viktor Kuncak

LARA - I&C - EPFL
emails: firstname.lastname@epfl.ch
INR 318, Station 15, CH-1015 Lausanne, Switzerland

Abstract. We present decision procedures for logical constraints involving collections such as sets, multisets, and fuzzy sets. Element membership in our collections is given by characteristic functions from a finite universe (of unknown size) to a user-defined subset of rational numbers. Our logic supports standard operators such as union, intersection, difference, or any operation defined pointwise using mixed linear integer-rational arithmetic. Moreover, it supports the notion of cardinality of the collection, defined as the sum of occurrences of all elements. Deciding formulas in such logic has applications in software verification.
Our decision procedure reduces satisfiability of formulas with collections to satisfiability of formulas in an extension of mixed linear integer-rational arithmetic with a "star" operator. The star operator computes the integer cone (closure under vector addition) of the solution set of a given formula. We give an algorithm for eliminating the star operator, which reduces the problem to mixed linear integer-rational arithmetic. Star elimination combines naturally with quantifier elimination for mixed integer-rational arithmetic. Our decidability result subsumes previous special cases for sets and multisets. The extension with star is interesting in its own right because it can encode reachability problems for a simple class of transition systems.

Keywords: verification and program analysis, sets, multisets, fuzzy sets, cardinality operator, mixed linear integer-rational arithmetic

1 Introduction

In this paper we show decidability of a logic for reasoning about collections of elements such as sets, multisets (bags), and fuzzy sets. We present a unified logic that can express all these kinds of collections and supports the cardinality operator on collections.

Our approach represents a collection of elements using its characteristic function $f : E \to R$. Inspired by applications in software verification [9], we assume that the domain E is a finite but of unknown size. The range R depends on the kind of the collection: for sets, $R = \{0, 1\}$; for multisets, $R = \{0, 1, 2, \ldots, \}$; for fuzzy sets, R is the interval $[0, 1]$ of rational numbers, denoted $\mathbb{Q}_{[0,1]}$. With this representation, operations and relations on collections such as union, difference, and subset are all expressed using operations of linear arithmetic. For example, the condition $A \cup B = C$ becomes $\forall e{\in}E.\ \max(A(e), B(e)) = C(e)$, a definition that applies whether A, B are sets, multisets, or fuzzy sets. A distinguishing feature of our constraints, compared to many other approaches for reasoning about functions $E \to R$, e.g. [2, Chapter 11], is the presence of the cardinality operator, defined by $|A| = \sum_{e \in E} A(e)$. The resulting language freely combines the uses linear arithmetic at two levels: the level of individual elements, as in the subformula $\max(A(e), B(e)) = C(e)$, and the level of sizes of collections, as in the formula $|A \cup B| + |A \cap B| = |A| + |B|$. The language subsumes constraints such as quantifier-free Boolean Algebra with Presburger Arithmetic [9] and therefore contains both set algebra and integer linear arithmetic. It also subsumes decidable constraints on multisets with cardinality bounds [12, 13].

The contribution of this paper is the decidability of constraints on collections where the range R is the set $\mathbb{Q}$ of all rational numbers. Our constraints can express the condition $(\forall e.int(A(e)) \wedge A(e) \geq 0)$ that the number of occurrences $A(e)$ for each element e is a non-negative integer number, so they subsume the case $R = \{0, 1, 2, \dots\}$ solved in [14, 13], which, in turn, subsumes the case of sets [9]. Moreover, our constraints can express the condition $\forall e.(0 \leq A(e) \leq 1)$, which makes them appropriate for modelling fuzzy sets.

Analogously to [12], our decision procedure is based on a translation of a formula with collections and cardinality constraints into a conjunction of a mixed linear integer-rational arithmetic (MLIRA) formula and a new form of condition, denoted $\boldsymbol{u} \in \{\boldsymbol{v} \mid F(\boldsymbol{v})\}^*$. Here the star operator denotes the integer conic hull of a set of rational vectors [5]. Therefore, $\{\boldsymbol{v} \mid F(\boldsymbol{v})\}^*$ denotes the closure under vector addition of the set of solution vectors $\boldsymbol{v}$ of the MLIRA formula F. Formally,

$$\boldsymbol{u} \in \{\boldsymbol{v} \mid F(\boldsymbol{v})\}^* \ \leftrightarrow\ \exists K \in \{0, 1, 2, \dots\}.\ \exists \boldsymbol{v}_1, \dots, \boldsymbol{v}_K.\ \boldsymbol{u} = \sum_{i=1}^{K} \boldsymbol{v}_i \wedge \bigwedge_{i=1}^{K} F(\boldsymbol{v}_i)$$

The star operator is interesting beyond its use in decidability of constraints on collections. For example, it can express the reachability condition for a transition system whose state is an integer or rational vector and whose transitions increment the vector by a solution of a given formula [13].

In contrast to the previous work [12, 13], the formula F in this paper is not restricted to integers, but can be arbitrary MLIRA formula. Consequently, we are faced with the problem of solving an extension of satisfiability of MLIRA formulas with the conditions $\boldsymbol{u} \in \{\boldsymbol{v} \mid F(\boldsymbol{v})\}^*$ where F is an arbitrary MLIRA formula. To solve this problem, we describe a finite and effectively computable representation of the solution set $S = \{\boldsymbol{v} \mid F(\boldsymbol{v})\}$. We use this representation to express the condition $\boldsymbol{u} \in S^*$ as a new MLIRA formula. This gives a "star elimination" algorithm. As one consequence, we obtain a unified decision procedure for sets, multisets, and fuzzy sets in the presence of the cardinality operator. As another consequence, we obtain the decidability of the extension of quantified mixed linear constraints [18] with stars.

Examples of constraints on sets. For each set variable s we assume the constraint $\forall e.(s(e) = 0 \vee s(e) = 1)$.

formula	informal description
$x \notin \texttt{content} \wedge \texttt{size} = \texttt{card content} \longrightarrow$ $(\texttt{size} = 0 \leftrightarrow \texttt{content} = \emptyset)$	using invariant on size to prove correctness of an efficient emptiness check
$x \notin \texttt{content} \wedge \texttt{size} = \texttt{card content} \longrightarrow$ $\texttt{size} + 1 = \texttt{card}(\{x\} \cup \texttt{content})$	maintaining correct size when inserting fresh element into set
$\texttt{size} = \texttt{card content} \wedge$ $\texttt{size1} = \texttt{card}(\{x\} \cup \texttt{content}) \longrightarrow$ $\texttt{size1} \leq \texttt{size} + 1$	maintaining size after inserting an element into set
$\texttt{content} \subseteq \texttt{alloc} \wedge$ $x_1 \notin \texttt{alloc} \wedge$ $x_2 \notin \texttt{alloc} \cup \{x_1\} \wedge$ $x_3 \notin \texttt{alloc} \cup \{x_1\} \cup \{x_2\} \longrightarrow$ $\texttt{card}\,(\texttt{content} \cup \{x_1\} \cup \{x_2\} \cup \{x_3\}) =$ $\texttt{card}\,\texttt{content} + 3$	allocating and inserting three objects into a container data structure
$\texttt{content} \subseteq \texttt{alloc0} \wedge x_1 \notin \texttt{alloc0} \wedge$ $\texttt{alloc0} \cup \{x_1\} \subseteq \texttt{alloc1} \wedge x_2 \notin \texttt{alloc1} \wedge$ $\texttt{alloc1} \cup \{x_2\} \subseteq \texttt{alloc2} \wedge x_3 \notin \texttt{alloc2} \longrightarrow$ $\texttt{card}\,(\texttt{content} \cup \{x_1\} \cup \{x_2\} \cup \{x_3\}) =$ $\texttt{card content} + 3$	allocating and inserting at least three objects into a container data structure
$x \in C \wedge C_1 = (C \setminus \{x\}) \wedge$ $\texttt{card}(\texttt{alloc1} \setminus \texttt{alloc0}) \leq 1 \wedge$ $\texttt{card}(\texttt{alloc2} \setminus \texttt{alloc1}) \leq \texttt{card}\, C_1 \longrightarrow$ $\texttt{card}\,(\texttt{alloc2} \setminus \texttt{alloc0}) \leq \texttt{card}\, C$	bound on the number of allocated objects in a recursive function that incorporates container C into another container

Examples of constraints on multisets. For each multiset variable m we assume the constraint $\forall e.int(m(e)) \wedge A(e) \geq 0$.

$\texttt{size} = \texttt{card content} \wedge$ $\texttt{size1} = \texttt{card}(\{x\} \uplus \texttt{content}) \longrightarrow$ $\texttt{size1} = \texttt{size} + 1$	maintaining size after inserting an element into multiset

Examples of constraints on fuzzy sets. For each fuzzy set variable f we assume the constraint $\forall e.0 \leq f(e) \leq 1$.

$2\vert A\vert \neq 2\vert B\vert + 1$	example formula valid over multisets but invalid over fuzzy sets
$(\forall e.U(e) = 1) \rightarrow \vert A \cap B\vert + \vert A \cup B\vert \leq \vert A\vert + \vert U\vert$	example formula valid over fuzzy sets but invalid over multisets
$(\forall e.C(e) = \lambda A(e) + (1 - \lambda)B(e)) \rightarrow$ $A \cap B \subseteq C \subseteq A \cup B$	basic property of convex combination of fuzzy sets [19], for any fixed constant $\lambda \in [0, 1]$

Fig. 1. Example constraints in our class.

2 Examples

Figure 1 shows small example formulas over sets, multisets, and fuzzy sets that are expressible in our logic. The examples for sets and multisets are based on verification conditions from software verification [9]. The remaining examples illustrate basic differences in valid formulas over multisets and fuzzy sets.

We illustrate our technique on one of the examples shown in Figure 1: we show that formula $\forall e.U(e) = 1 \rightarrow |A \cap B| + |A \cup B| \leq |A| + |U|$ is valid where U, A, and B are fuzzy sets. To prove formula validity, we prove unsatisfiability of its negation, conjoined with the constraints ensuring that the collections are fuzzy sets:

$$\begin{array}{l}\forall e.U(e) = 1 \ \wedge \ |A| + |U| < |A \cap B| + |A \cup B| \ \wedge \\ \forall e.0 \leq A(e) \leq 1 \ \wedge \ \forall e.0 \leq B(e) \leq 1 \ \wedge \ \forall e.0 \leq U(e) \leq 1\end{array}$$

We first reduce the formula to the normal form, as follows. We flatten the formula by introducing fresh variables n_i for each cardinality operator. The formula reduces to:

$$\begin{array}{l}n_1 + n_2 < n_3 + n_4 \ \wedge \ n_1 = |A| \ \wedge \ n_2 = |U| \ \wedge \ n_3 = |A \cap B| \ \wedge \ n_4 = |A \cup B| \ \wedge \\ \forall e.U(e) = 1 \ \wedge \ \forall e.0 \leq A(e) \leq 1 \ \wedge \ \forall e.0 \leq B(e) \leq 1 \ \wedge \ \forall e.0 \leq U(e) \leq 1\end{array}$$

We next apply the definition of the cardinality operator, $|C| = \sum_{e \in E} C(e)$:

$$\begin{array}{l}n_1 + n_2 < n_3 + n_4 \ \wedge \ n_1 = \sum_{e \in E} A(e) \ \wedge \ n_2 = \sum_{e \in E} U(e) \ \wedge \\ n_3 = \sum_{e \in E} (A \cap B)(e) \ \wedge \ n_4 = \sum_{e \in E} (A \cup B)(e) \ \wedge \\ \forall e.U(e) = 1 \ \wedge \ \forall e.0 \leq A(e) \leq 1 \ \wedge \ \forall e.0 \leq B(e) \leq 1 \ \wedge \ \forall e.0 \leq U(e) \leq 1\end{array}$$

Operators $\cup$ and $\cap$ are defined pointwise using ite operator:

$$(C_1 \cup C_2)(e) = \max\{C_1(e), C_2(e)\} = \mathsf{ite}(C_1(e) \leq C_2(e), C_2(e), C_1(e))$$

$$(C_1 \cap C_2)(e) = \min\{C_1(e), C_2(e)\} = \mathsf{ite}(C_1(e) \leq C_2(e), C_1(e), C_2(e)),$$

where $\mathsf{ite}(A, B, C)$ is the standard *if-then-else* operator, denoting B when A is true and C otherwise. Using these definitions, the example formula becomes:

$$\begin{array}{c}n_1 + n_2 < n_3 + n_4 \ \wedge \ n_1 = \sum\limits_{e \in E} A(e) \ \wedge \ n_2 = \sum\limits_{e \in E} U(e) \ \wedge \\ n_3 = \sum\limits_{e \in E} \mathsf{ite}(A(e) \leq B(e), A(e), B(e)) \wedge n_4 = \sum\limits_{e \in E} \mathsf{ite}(A(e) \leq B(e), B(e), A(e)) \wedge \\ \forall e.U(e) = 1 \ \wedge \ \forall e.0 \leq A(e) \leq 1 \ \wedge \ \forall e.0 \leq B(e) \leq 1 \ \wedge \ \forall e.0 \leq U(e) \leq 1\end{array}$$

Using vectors of integers, we then group all the sums into one, and also group all universally quantified constraints:

$$\begin{array}{l}n_1 + n_2 < n_3 + n_4 \wedge \big(n_1, n_2, n_3, n_4\big) = \\ \sum\limits_{e \in E} \big(A(e),\ U(e),\ \mathsf{ite}(A(e) \leq B(e), A(e), B(e)),\ \mathsf{ite}(A(e) \leq B(e), B(e), A(e))\big) \\ \qquad \wedge\ \forall e.\ \big(U(e) = 1 \ \wedge \ 0 \leq A(e) \leq 1 \ \wedge \ 0 \leq B(e) \leq 1 \ \wedge \ 0 \leq U(e) \leq 1\big)\end{array}$$

As we prove in Theorem 1 below, the last formula is equisatisfiable with

$$n_1 + n_2 < n_3 + n_4 \land (n_1, n_2, n_3, n_4) \in \\ \{\big(a, u, \mathsf{ite}(a \le b, a, b), \mathsf{ite}(a \le b, b, a)\big) \mid u = 1 \;\land\; 0 \le a \le 1 \;\land\; 0 \le b \le 1\}^*$$

The subject of this paper are general techniques for solving such satisfiability problems that contain a MLIRA formula and a star operator applied to another MLIRA formula. We next illustrate some of the ideas of the general technique, taking several shortcuts to keep the exposition brief.

Because the value of the variable u is determined ($u = 1$), we can simplify the last formula to:

$$n_1 + n_2 < n_3 + n_4 \;\land\; (n_1, n_2, n_3, n_4) \in S^*$$

where $S = \{\big(a, 1, \mathsf{ite}(a \le b, a, b), \mathsf{ite}(a \le b, b, a)\big) \mid 0 \le a \le 1 \land 0 \le b \le 1\}$. By case analysis on $a \le b$, we conclude $S = S_1 \cup S_2$ for

$$S_1 = \{(a, 1, a, b) \mid 0 \le a \le 1 \land 0 \le b \le 1 \land a \le b\}$$
$$S_2 = \{(a, 1, b, a) \mid 0 \le a \le 1 \land 0 \le b \le 1 \land b < a\}$$

This eliminates the ite expressions and we have:

$$n_1 + n_2 < n_3 + n_4 \land (n_1, n_2, n_3, n_4) \in (S_1 \cup S_2)^*$$

By definition of star operator, the last condition is equivalent to

$$n_1 + n_2 < n_3 + n_4 \;\land\; (n_1, n_2, n_3, n_4) = (n_1^1, n_2^1, n_3^1, n_4^1) + (n_1^2, n_2^2, n_3^2, n_4^2) \;\land \\ (n_1^1, n_2^1, n_3^1, n_4^1) \in S_1^* \;\land\; (n_1^2, n_2^2, n_3^2, n_4^2) \in S_2^*$$

Let us characterize the condition $(n_1^1, n_2^1, n_3^1, n_4^1) \in S_1^*$. Let K_1 denote the number of vectors in S_1 whose sum is $(n_1^1, n_2^1, n_3^1, n_4^1)$. By definition of the star operator, there are $a_1^1, \ldots, a_{K_1}^1$ and $b_1^1, \ldots, b_{K_1}^1$ such that $0 \le a_i^1 \le b_i^1 \le 1$ and

$$(n_1^1, n_2^1, n_3^1, n_4^1) = \sum_{i=1}^{K_1} (a_i^1, 1, a_i^1, b_i^1)$$

We obtain that $n_1^1 = n_3^1 = \sum_{i=1}^{K_1} a_i^1 = A_1$, $n_2 = K_1$, $n_4 = \sum_{i=1}^{K_1} b_i^1 = B_1$. The other case for S_2 is analogous and we derive $(n_1^2, n_2^2, n_3^2, n_4^2) = (A_2, K_2, B_2, A_2)$.

This way we eliminate the star operator and the example formula becomes:

$$n_1 + n_2 < n_3 + n_4 \land (n_1, n_2, n_3, n_4) = (A_1, K_1, A_1, B_1) + (A_2, K_2, B_2, A_2)$$

This formula further reduces to $K_1 + K_2 < B_1 + B_2$. If we apply the definitions of B_i and properties of b_i^j we obtain the following formula:

$$K_1 + K_2 < \sum_{i=1}^{K_1} b_i^1 + \sum_{i=1}^{K_2} b_i^2 \;\land\; \bigwedge_{i=1}^{K_1} b_i^1 \le 1 \land \bigwedge_{i=1}^{K_2} b_i^2 \le 1$$

In this case, it is easy to see that the resulting formula is contradictory. This shows that the initial formula is valid over fuzzy sets. Our paper shows that, in general, such formulas are equivalent to existentially quantified MLIRA formulas, despite the fact that their initial formulation involves sums with parameters such as K_1 and K_2. This is possible thanks to the special structure of the sets of solutions of MLIRA formulas, which we describe building on results such as [7] and the theory of linear programming.

Having seen the use of our method to prove formula validity, we illustrate its use in producing counterexamples by showing that the original formula is invalid over multisets. Restricting the range of each collection to integers and using the same reduction, we derive formula

$$\begin{aligned}&n_1 + n_2 < n_3 + n_4 \wedge \\ &(n_1, n_2, n_3, n_4) \in \{\big(a, 1, \mathsf{ite}(a \le b, a, b), \mathsf{ite}(a \le b, b, a)\big) \mid a, b \in \mathbb{N}\}^*\end{aligned}$$

Applying again a similar case analysis, we deduce $K_1 + K_2 < \sum_{i=1}^{K_1} b_i^1 + \sum_{i=1}^{K_2} b_i^2$ where all b_i^j's are non-negative integers. This formula is satisfiable, for example, with a satisfying variable assignment $K_1 = 1, b_1^1 = 2$ and $K_2 = 0$. The correspondence of Theorem 1 then allows us to construct a multiset counterexample. Because $K_2 = 0$, no vector from S_2 contributes to sum and we consider only S_1. Variable K_1 denotes the number of elements of a domain set E, so we consider the domain set $E = \{e_1\}$. Multisets A, B and U are defined by $A(e_1) = 1$, $B(e_1) = 2$, and $U(e_1) = 1$. It can easily be verified that this is a counterexample for validity of the formula over multisets.

3 From Collections to Stars

This section describes the translation from constraints on collections to constraints that use star operator. We first present the syntax of our constraints and clarify the semantics of selected constructs (the semantics of the remaining constructs can be derived from their translation into simpler ones).

We model each collection f as a function whose domain is a finite set E of unknown size and whose range is the set of rational numbers. When the constraints imply that the range of f is $\{0, 1\}$, then f models sets, when the range of f are non-negative integers, then f denotes standard multisets (bags), in which an element can occur multiple times. We call the number of occurrences of an element e, denoted $f(e)$, the *multiplicity* of an element. When the range of f is restricted to be in interval $[0, 1]$, then f describes a fuzzy set [19].

In addition to standard operations on collections (such as plus, union, intersection, difference) we also allow the cardinality operator, defined as $|f| = \sum_{e \in E} f(e)$. This is the desired definition for sets and multisets and we believe it is a natural notion for fuzzy sets over a finite universe E. Figure 2 shows a context-free grammar of our formulas involving collections.

Semantics of some less commonly known operators is defined as follows: $\mathsf{ite}(A, B, C)$ denotes the if-then-else expression, which evaluates to B when A is top-level formulas:

$F ::= A \mid F \wedge F \mid \neg F$

$A ::= C{=}C \mid C \subseteq C \mid \forall e.\mathsf{F}^{\mathsf{in}} \mid \mathsf{A}^{\mathsf{out}}$

outer linear arithmetic formulas:

$\mathsf{F}^{\mathsf{out}} ::= \mathsf{A}^{\mathsf{out}} \mid \mathsf{F}^{\mathsf{out}} \wedge \mathsf{F}^{\mathsf{out}} \mid \neg \mathsf{F}^{\mathsf{out}}$

$\mathsf{A}^{\mathsf{out}} ::= \mathsf{t}^{\mathsf{out}} \leq \mathsf{t}^{\mathsf{out}} \mid \mathsf{t}^{\mathsf{out}}{=}\mathsf{t}^{\mathsf{out}} \mid (\mathsf{t}^{\mathsf{out}}, \ldots, \mathsf{t}^{\mathsf{out}}){=}\sum_{\mathsf{F}^{\mathsf{in}}}(\mathsf{t}^{\mathsf{in}}, \ldots, \mathsf{t}^{\mathsf{in}})$

$\mathsf{t}^{\mathsf{out}} ::= k \mid \; |C| \; \mid K \mid \mathsf{t}^{\mathsf{out}} + \mathsf{t}^{\mathsf{out}} \mid K \cdot \mathsf{t}^{\mathsf{out}} \mid \lfloor \mathsf{t}^{\mathsf{out}} \rfloor \mid \mathsf{ite}(\mathsf{F}^{\mathsf{out}}, \mathsf{t}^{\mathsf{out}}, \mathsf{t}^{\mathsf{out}})$

inner linear arithmetic formulas:

$\mathsf{F}^{\mathsf{in}} ::= \mathsf{A}^{\mathsf{in}} \mid \mathsf{F}^{\mathsf{in}} \wedge \mathsf{F}^{\mathsf{in}} \mid \neg \mathsf{F}^{\mathsf{in}}$

$\mathsf{A}^{\mathsf{in}} ::= \mathsf{t}^{\mathsf{in}} \leq \mathsf{t}^{\mathsf{in}} \mid \mathsf{t}^{\mathsf{in}}{=}\mathsf{t}^{\mathsf{in}}$

$\mathsf{t}^{\mathsf{in}} ::= f(e) \mid K \mid \mathsf{t}^{\mathsf{in}} + \mathsf{t}^{\mathsf{in}} \mid K \cdot \mathsf{t}^{\mathsf{in}} \mid \lfloor \mathsf{t}^{\mathsf{in}} \rfloor \mid \mathsf{ite}(\mathsf{F}^{\mathsf{in}}, \mathsf{t}^{\mathsf{in}}, \mathsf{t}^{\mathsf{in}})$

expressions about collections:

$C ::= c \mid \emptyset \mid C \cap C \mid C \cup C \mid C \uplus C \mid C \setminus C \mid C \setminus\setminus C \mid \mathsf{setof}(C)$

terminals:

c - collection variable; e - index variable (fixed)

k - rational variable; K - rational constant

Fig. 2. Quantifier-Free Formulas about Collection with Cardinality Operator

true and evaluates to C when A is false. The $\mathsf{setof}(C)$ operator takes as an argument collection C and returns the set of all elements for which $C(e)$ is positive. To constrain a variable s to denote a set, use formula $\forall e. s(e) = 0 \vee s(e) = 1$. To constraint a variable m to denote a multiset, use formula $(\forall e. int(m(e)) \wedge m(e) \geq 0)$. Here $int(x)$ is a shorthand for $\lfloor x \rfloor = x$ where $\lfloor x \rfloor$ is the largest integer smaller than or equal to x.

A decision procedure for checking satisfiability of the subclass of integer formulas was described in [12]. The novelty of constraints in Figure 2 compared to the language in [12] is the presence of the floor operator $\lfloor x \rfloor$ and not only integer but also rational constants. All variables in our current language are interpreted over rationals, but any of them can be restricted to be integer using the constraint $int(x)$.

To reduce reasoning about collections to reasoning in linear arithmetic with stars, we follow the idea from [12] and convert a formula to the *sum normal form*.

Definition 1. *A formula is in sum normal form iff it is of the form*

$$P \;\wedge\; (u_1, \ldots, u_n) = \sum_{e \in E}(t_1, \ldots, t_n) \;\wedge\; \forall e.F$$

where P is a quantifier-free linear arithmetic formula with no collection variables, and where variables in $t_1, \ldots, t_n$ and F occur only as expressions of the form $c(e)$ for a collection variable c and e the fixed index variable.

Figure 3 summarizes the process of transforming formula into sum normal form.[1] The previous example section illustrated this idea. As another example, consider a negation of a formula that verifies the change in the size of a list after insertion of an element: $|x| = 1 \wedge |L \uplus x| \neq |L| + 1$. The sum normal form of this formula is: $k_1 \neq k_2 + 1 \;\wedge\; (1, k_1, k_2) = \sum_{e \in E}(x(e), y(e), L(e)) \;\wedge\; \forall e. y(e) = L(e) + x(e)$.

INPUT: formula in the syntax of Figure 2
OUTPUT: formula in sum normal form (Definition 1)

1. Flatten expressions that we wish to eliminate:
   $C[exp] \rightsquigarrow (x = exp \wedge C[x])$
   where exp is one of the expressions $\emptyset$, $c_1 \cup c_2$, $c_1 \cap c_2$, $c_1 \uplus c_2$, $c_1 \setminus c_2$, $\mathsf{setof}(c_1)$, $|c_1|$, and where the occurrence of exp is not already in a top-level conjunct of the form $x = exp$ or $exp = x$ for some variable x.
2. Reduce colection relations to pointwise linear arithmetic conditions:
   $$\begin{array}{ll} C[c_0 = \emptyset] & \rightsquigarrow C[\forall e.\ c_0(e) = 0] \\ C[c_0 = c_1 \cap c_2] & \rightsquigarrow C[\forall e.\ c_0(e) = \mathsf{ite}(c_1(e) \leq c_2(e), c_1(e), c_2(e))] \\ C[c_0 = c_1 \cup c_2] & \rightsquigarrow C[\forall e.\ c_0(e) = \mathsf{ite}(c_1(e) \leq c_2(e), c_2(e), c_1(e))] \\ C[c_0 = c_1 \uplus c_2] & \rightsquigarrow C[\forall e.\ c_0(e) = c_1(e) + c_2(e)] \\ C[c_0 = c_1 \setminus c_2] & \rightsquigarrow C[\forall e.\ c_0(e) = \mathsf{ite}(c_1(e) \leq c_2(e), 0, c_1(e) - c_2(e))] \\ C[c_0 = c_1 \setminus\!\!\setminus c_2] & \rightsquigarrow C[\forall e.\ c_0(e) = \mathsf{ite}(c_2(e) = 0, c_1(e), 0)] \\ C[c_0 = \mathsf{setof}(c_1)] & \rightsquigarrow C[\forall e.\ c_0(e) = \mathsf{ite}(0 < c_1(e), 1, 0)] \\ C[c_1 \subseteq c_2] & \rightsquigarrow C[\forall e.\ (c_1(e) \leq c_2(e))] \\ C[c_1 = c_2] & \rightsquigarrow C[\forall e.\ (c_1(e) = c_2(e))] \end{array}$$
3. Express each cardinality operator using a sum:
   $$C[\,|c|\,] \rightsquigarrow C[\sum_{e \in E} c(e)]$$
4. Express negatively occurring pointwise definitions using the sum:
   $$C[\forall e.F] \rightsquigarrow C[0 = \sum_{e \in E} \mathsf{ite}(F(e), 0, 1)]$$
5. Flatten any sums that are not already top-level conjuncts:
   $$C[\,(u_1, \ldots, u_n) {=} \sum_F (t_1, \ldots, t_n)\,] \;\rightsquigarrow\; (w_1, \ldots, w_n) {=} \sum_F (t_1, \ldots, t_n) \wedge C[\bigwedge_{i=1}^{n} u_i {=} w_i\,]$$
6. Eliminate conditions from sums:
   $$C[\sum_F (t_1, \ldots, t_n)\,] \rightsquigarrow C[\sum_{e \in E} (\mathsf{ite}(F, t_1, 0), \ldots, \mathsf{ite}(F, t_n, 0))\,]$$
7. Group all sums into one:
   $$P \wedge \bigwedge_{i=1}^{q} (u_1^i, \ldots, u_{n_i}^i) = \sum_{e \in E} (t_1^i, \ldots, t_{n_i}^i) \;\rightsquigarrow$$
   $$P \wedge (u_1^1, \ldots, u_{n_1}^1, \ldots, u_1^q, \ldots, u_{n_q}^q) = \sum_{e \in E} (t_1^1, \ldots, t_{n_1}^1, \ldots, t_1^q, \ldots, t_{n_q}^q)$$
8. Group all pointwise defined operations into one:
   $$P \wedge \bigwedge_{i=1}^{q} (\forall e.F_i) \;\rightsquigarrow\; P \wedge \forall e. \bigwedge_{i=1}^{q} F_i$$

Fig. 3. Algorithm for reducing collections formulas to sum normal form

[1] Note that the part $\forall e.F$ could be omitted from normal form definition and expressed as an additional component of the sum. However, its use leads to somewhat simpler constraints.

Formulas in sum normal form contain only one top-level sum which ranges over elements of an existentially quantified set E. To study such constraints we introduce the star operator.

Definition 2 (Star operator, integer conic hull [5]). *Let C be a set of rational vectors. Define $C^* = \{\boldsymbol{v}_1 + \ldots + \boldsymbol{v}_K \mid K \in \{0,1,2,\ldots\}, \boldsymbol{v}_1,\ldots,\boldsymbol{v}_K \in C\}$.*

The fact that the bound variable K in Definition 2 ranges over non-negative integers as opposed to rational or real numbers differentiates the integer conic hull (star) from the notion of conic hull in linear programming [17].

Theorem 1. *A formula $(u_1,\ldots,u_n) = \sum_{e\in E}(t_1,\ldots,t_n) \;\wedge\; \forall e.F$ is equisatisfiable with the formula $(u_1,\ldots,u_n) \in \{(t'_1,\ldots,t'_n) \mid x_i \in \mathbb{Q} \wedge F'\}^*$ where t'_j and F' are t_j and F respectively in which each $c_i(e)$ is replaced by a fresh variable x_i.*

Proof. $\Leftarrow$): Assume $(u_1,\ldots,u_n) \in \{(t'_1,\ldots,t'_n) \mid x_i \in \mathbb{Q} \wedge F'\}^*$ is satisfiable. Then there exists an integer $k \geq 0$ such that $(u_1,\ldots,u_n) = \sum_{j=1}^{k}(t^j_1,\ldots,t^j_n)$. We define set E to consist of k distinct elements, $E = \{e_1,\ldots,e_k\}$. Every variable x_i occurring in $t'_1,\ldots,t'_n$ and F' corresponds to the collection c_i. Let x^j_i denote the value of x_i in jth summand $(t^j_1,\ldots,t^j_n)$. Define each collection c_i by $c_i(e_j) = x^j_i$. The finite set E and collections c_i defined as above make formula $(u_1,\ldots,u_n) = \sum_{e\in E}(t_1,\ldots,t_n) \;\wedge\; \forall e.F$ satisfiable.

$\Rightarrow$): The other direction is analogous. Given E, for each $e_j \in E$ we obtain a set of values $c_i(e_j)$ that give the values for x_i in jth summand. ■

Applying Theorem 1 to our example of insertion into a list, we obtain that

$$k_1 \neq k_2 + 1 \;\wedge\; (1, k_1, k_2) = \sum_{e\in E}(x(e), y(e), L(e)) \;\wedge\; \forall e. y(e) = L(e) + x(e)$$

is equisatisfiable with

$$k_1 \neq k_2 + 1 \;\wedge\; (1, k_1, k_2) \in \{(x, y, L) \mid y = L + x\}^*$$

Thanks to Theorem 1, in the rest of the paper we investigate the satisfiability problem for such formulas, whose syntax is given in Figure 4. These formulas are sufficient to check satisfiability for formulas in Figure 2. In Section 6 we present a more general decidable language that allows nesting of terms, logical operations, quantifiers, and stars.

4 Separating Mixed Constraints

As justified in previous sections, we consider the satisfiability problem for $G(\boldsymbol{r},\boldsymbol{w}) \wedge \boldsymbol{w} \in \{\boldsymbol{x} \mid F(\boldsymbol{x})\}^*$ where F and G are quantifier-free, mixed linear integer-rational arithmetic (MLIRA) formulas.

Our goal is to give an algorithm for constructing another MLIRA formula F' such that $\boldsymbol{w} \in \{\boldsymbol{x} \mid F(\boldsymbol{x})\}^*$ is equivalent to $\exists \boldsymbol{w}'. F'(\boldsymbol{w}',\boldsymbol{w})$. This will reduce the satisfiability problem to the satisfiability of $G(\boldsymbol{r},\boldsymbol{w}) \wedge F'(\boldsymbol{w}',\boldsymbol{w})$.

As a first stage towards this goal, this section shows how to represent the set $\{\boldsymbol{x} \mid F(\boldsymbol{x})\}$ using solutions of pure integer constraints and solutions of pure rational constraints. We proceed in several steps.

top-level, outer linear arithmetic formulas:
$\mathsf{F}^{\mathsf{out}} ::= \mathsf{A}^{\mathsf{out}} \mid \mathsf{F}^{\mathsf{out}} \wedge \mathsf{F}^{\mathsf{out}} \mid \neg \mathsf{F}^{\mathsf{out}}$
$\mathsf{A}^{\mathsf{out}} ::= \mathsf{t}^{\mathsf{out}} \leq \mathsf{t}^{\mathsf{out}} \mid \mathsf{t}^{\mathsf{out}} = \mathsf{t}^{\mathsf{out}} \mid (\mathsf{t}^{\mathsf{out}}, \ldots, \mathsf{t}^{\mathsf{out}}) \in \{(\mathsf{t}^{\mathsf{in}}, \ldots, \mathsf{t}^{\mathsf{in}}) \mid \mathsf{F}^{\mathsf{in}}\}^*$
$\mathsf{t}^{\mathsf{out}} ::= \mathsf{k}^{\mathsf{out}} \mid K \mid \mathsf{t}^{\mathsf{out}} + \mathsf{t}^{\mathsf{out}} \mid K \cdot \mathsf{t}^{\mathsf{out}} \mid \lfloor \mathsf{t}^{\mathsf{out}} \rfloor \mid \mathsf{ite}(\mathsf{F}^{\mathsf{out}}, \mathsf{t}^{\mathsf{out}}, \mathsf{t}^{\mathsf{out}})$
inner linear arithmetic formulas:
$\mathsf{F}^{\mathsf{in}} ::= \mathsf{A}^{\mathsf{in}} \mid \mathsf{F}^{\mathsf{in}} \wedge \mathsf{F}^{\mathsf{in}} \mid \neg \mathsf{F}^{\mathsf{in}}$
$\mathsf{A}^{\mathsf{in}} ::= \mathsf{t}^{\mathsf{in}} \leq \mathsf{t}^{\mathsf{in}} \mid \mathsf{t}^{\mathsf{in}} = \mathsf{t}^{\mathsf{in}}$
$\mathsf{t}^{\mathsf{in}} ::= \mathsf{k}^{\mathsf{in}} \mid K \mid \mathsf{t}^{\mathsf{in}} + \mathsf{t}^{\mathsf{in}} \mid K \cdot \mathsf{t}^{\mathsf{in}} \mid \lfloor \mathsf{t}^{\mathsf{in}} \rfloor \mid \mathsf{ite}(\mathsf{F}^{\mathsf{in}}, \mathsf{t}^{\mathsf{in}}, \mathsf{t}^{\mathsf{in}})$
terminals:
$\mathsf{k}^{\mathsf{in}}, \mathsf{k}^{\mathsf{out}}$ - rational variable (two disjoint sets); K - rational constants

Fig. 4. Syntax of Mixed Integer-Rational Linear Arithmetic with Star

Step 1. Eliminate the floor functions from F using integer and real variables, applying from left to right the equivalence

$$C(\lfloor t \rfloor) \leftrightarrow \exists y_Q \in \mathbb{Q}. \exists y_Z \in \mathbb{Z}.\ t = y_Q \wedge y_Z \leq y_Q < y_Z + 1 \wedge C(y_Z)$$

The result is an equivalent formula without the floor operators, where some of the variables are restricted to be integer.

Step 2. Transform F into linear programming problems, as follows. First, eliminate if-then-else expressions by introducing fresh variables and using disjunction (see e.g. [12]). Then transform formula to negation normal form. Eliminate $t_1 = t_2$ by transforming it into $t_1 \leq t_2 \wedge t_2 \leq t_1$. Eliminate $t_1 \neq t_2$ by transforming it into $t_1 < t_2 \vee t_2 < t_1$. Following [4, Section 3.3], replace each $t_1 < t_2$ with $t_1 + \delta \leq t_2$ where δ is a special variable (the same for all strict inequalities), for which we require $0 < \delta \leq 1$. We obtain for some d matrices A_i for $1 \leq i \leq d$ such that

$$F(\boldsymbol{x}) \ \leftrightarrow\ \exists \boldsymbol{y}^Z \in \mathbb{Z}^{d_Z}. \exists \boldsymbol{y}^Q \in \mathbb{Q}^{d_Q}. \exists \delta \in \mathbb{Q}_{(0,1]}. \bigvee_{i=1}^{d} A_i \cdot (\boldsymbol{x}, \boldsymbol{y}^Z, \boldsymbol{y}^Q) \leq \boldsymbol{b}$$

where $A_i \cdot (\boldsymbol{x}, \boldsymbol{y}^Z, \boldsymbol{y}^Q)$ denotes multiplication of matrix A_i by the vector $(\boldsymbol{x}, \boldsymbol{y}^Z, \boldsymbol{y}^Q)$ obtained by stacking vectors $\boldsymbol{x}$, $\boldsymbol{y}^Z$, and $\boldsymbol{y}^Q$.

Step 3. Represent the rational variables $\boldsymbol{x}$, $\boldsymbol{y}^Q$ as a sum of its integer part and its fractional part from $\mathbb{Q}_{[0,1]}$, obtaining

$$F(\boldsymbol{x}) \ \leftrightarrow\ \Big(\exists (\boldsymbol{x}^Z, \boldsymbol{y}^Z) \in \mathbb{Z}^{d'_Z}. \exists (\boldsymbol{x}^R, \boldsymbol{y}^R) \in \mathbb{Q}_{[0,1]}^{d'_Q}. \exists \delta \in \mathbb{Q}_{(0,1]}. \\ \boldsymbol{x} = \boldsymbol{x}^Z + \boldsymbol{x}^R \ \wedge\ \bigvee_{i=1}^{d} A'_i \cdot (\boldsymbol{x}^Z, \boldsymbol{y}^Z, \boldsymbol{x}^R, \boldsymbol{y}^R) \leq \boldsymbol{b}' \Big)$$

Note that $\boldsymbol{w} \in \{\boldsymbol{x} \mid \exists \boldsymbol{y}. H(\boldsymbol{x}, \boldsymbol{y})\}^*$ is equivalent to

$$\exists \boldsymbol{w}'. (\boldsymbol{w}, \boldsymbol{w}') \in \{(\boldsymbol{x}, \boldsymbol{y}) \mid H(\boldsymbol{x}, \boldsymbol{y})\}^*$$

In other words, we can push existential quantifiers to the top-level of the formula. Therefore, the original problem (after renaming) becomes

$$G(\boldsymbol{r},\boldsymbol{w}) \ \wedge\ \exists \boldsymbol{z}.\ (\boldsymbol{u}^Z,\boldsymbol{u}^Q,\varDelta)\ \in$$
$$\{(\boldsymbol{x}^Z,\boldsymbol{x}^R,\delta) \mid \bigvee_{i=1}^{d} A_i\cdot(\boldsymbol{x}^Z,\boldsymbol{x}^R,\delta) \le \boldsymbol{b}_i,\ \boldsymbol{x}^Z \in \mathbb{Z}^{d_Z}, \boldsymbol{x}^R \in \mathbb{Q}^{d_R}_{[0,1]}, \delta \in \mathbb{Q}_{(0,1]}\}^*$$

where the vector $\boldsymbol{z}$ contains a subset of variables $\boldsymbol{u}^Z,\boldsymbol{u}^Q,\varDelta$.

Step 4. Separate integer and rational parts, as follows. Consider one of the disjuncts $A\cdot(\boldsymbol{x}^Z,\boldsymbol{y}^R,\delta) \le \boldsymbol{b}$. For $A = [A_Z\ A_R\ c]$ this linear condition can be written as $A_Z\boldsymbol{x}^Z + A_R\boldsymbol{x}^R + \boldsymbol{c}\delta \le \boldsymbol{b}$, that is

$$A_R\boldsymbol{x}^R + \boldsymbol{c}\delta \le \boldsymbol{b} - A_Z\boldsymbol{x}^Z \tag{1}$$

Because the right-hand side is integer, for $\boldsymbol{a}$ denoting $\lceil A_R\boldsymbol{x}^R + \boldsymbol{c}\delta\rceil$ (left-hand side rounded up), the equation becomes $A_R\boldsymbol{x}^R + \boldsymbol{c}\delta \le \boldsymbol{a} \le \boldsymbol{b} - A_Z\boldsymbol{x}^Z$. Because $\boldsymbol{x}^R \in \mathbb{Q}^{d_Q}_{[0,1]}, \delta \in \mathbb{Q}_{(0,1]}$, vector $\boldsymbol{a}$ is bounded by the norm M_1 of the matrix $[A_R\ \boldsymbol{c}]$. Formula (1) is therefore equivalent to the finite disjunction

$$\bigvee_{\boldsymbol{a}\in\mathbb{Z}^d, ||\boldsymbol{a}||\le M_1} A_Z\boldsymbol{x}^Z \le \boldsymbol{b} - \boldsymbol{a} \wedge A_R\boldsymbol{x}^R + \boldsymbol{c}\delta \le \boldsymbol{a} \tag{2}$$

Note that each disjunct is a conjunction of a purely integer constraint and a purely rational constraint.

Step 5. Propagate star through disjunction, using the property

$$\boldsymbol{w} \in \{\boldsymbol{x} \mid \bigvee_{i=1}^{n} H_i(\boldsymbol{x})\}^* \ \leftrightarrow\ \exists \boldsymbol{w}_1,\ldots,\boldsymbol{w}_n.\ \boldsymbol{w} = \sum_{i=1}^{n}\boldsymbol{w}_i \ \wedge\ \bigwedge_{i=1}^{n} \boldsymbol{w}_i \in \{\boldsymbol{x} \mid H_i(\boldsymbol{x})\}^*$$

The final result is an equivalent conjunction of a MLIRA formula and an existentially quantified conjunction of formulas of the form

$$\begin{aligned}(\boldsymbol{u}^Z,\boldsymbol{u}^Q,\varDelta) \in \{(\boldsymbol{x}^Z,\boldsymbol{x}^R,\delta) \mid\ & A_Z\boldsymbol{x}^Z \le \boldsymbol{b}_Z, A_R\cdot(\boldsymbol{x}^R,\delta) \le \boldsymbol{b}_R,\\ & \boldsymbol{x}^Z \in \mathbb{Z}^{d_Z}, \boldsymbol{x}^R \in \mathbb{Q}^{d_R}_{[0,1]}, \delta \in \mathbb{Q}_{(0,1]}\}^*\end{aligned} \tag{3}$$

5 Eliminating Star Operator from Formulas

The previous section sets the stage for the following star-elimination theorem, which is the core result of this paper.

Theorem 2. *Let F be a quantifier-free MLIRA formula. Then there exist effectively computable integer vectors $\boldsymbol{a}_i$ and $\boldsymbol{b}_{ij}$ and effectively computable rational vectors $\boldsymbol{c}_1,\ldots,\boldsymbol{c}_n$ with coordinates in $\mathbb{Q}_{[0,1]}$ such that formula (3) is equivalent* *to a formula of the form*

$$
\begin{aligned}
&\exists K \in \mathbb{N}.\ \exists \mu_1, \ldots, \mu_q, \nu_{11}, \ldots, \nu_{qq_q} \in \mathbb{N}.\ \exists \beta_1, \ldots, \beta_n \in \mathbb{Q}. \\
&\big(\boldsymbol{u}^Z = \sum_{i=1}^{q} (\mu_i \boldsymbol{a}_i + \sum_{j=1}^{q_i} \nu_{ij} \boldsymbol{b}_{ij}) \ \wedge \bigwedge_{i=1}^{q} (\mu_i = 0 \to \sum_{j=1}^{q_i} \nu_{ij} = 0) \ \wedge \ (\sum_{i=1}^{q} \mu_i = K)\big) \\
&\qquad\qquad \wedge \quad \big((K = 0 \wedge \Delta = 0 \wedge \boldsymbol{u}^Q = \boldsymbol{0}) \vee \\
&\quad (K \geq 1 \wedge \Delta > 0 \wedge (\boldsymbol{u}^Q, \Delta) = \sum_{i=1}^{n} \beta_i \boldsymbol{c}_i \ \wedge \ \bigwedge_{i=1}^{n} \beta_i \geq 0 \ \wedge \ \sum_{i=1}^{n} \beta_i = K)\big)
\end{aligned} \tag{4}
$$

Proof. For a set of vectors S and an integer variable K, we define $KS = \{v_1 + \ldots + v_K \mid v_1, \ldots, v_K \in S\}$. Formula (3) is satisfiable iff there exists non-negative integer $K \in \mathbb{N}$ such that both

$$\boldsymbol{u}^Z \in K\{\boldsymbol{x}^Z \mid A_Z \boldsymbol{x}^Z \leq \boldsymbol{b}^Z\} \tag{5}$$

and

$$(\boldsymbol{u}^Q, \Delta) \in K\{(\boldsymbol{x}^R, \delta) \mid A_R \cdot (\boldsymbol{x}^R, \delta) \leq \boldsymbol{b}^R, \boldsymbol{x}^R \in \mathbb{Q}_{[0,1]}^{d_R}, \delta \in \mathbb{Q}_{(0,1]}\} \tag{6}$$

hold. We show how to describe (5) and (6) as existentially quantified MLIRA formulas that share the variable K.

To express formula (5) as a MLIRA formula, we use the fact that solutions of integer linear arithmetic formulas are semilinear sets (see [7], [11, Proposition 2]). Semilinear sets are finite unions of sets of a form $\{\boldsymbol{a}\} + \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n\}^*$. A sum of two sets is the Minkowski sum: $A + B = \{\boldsymbol{a} + \boldsymbol{b} \mid \boldsymbol{a} \in A,\ \boldsymbol{b} \in B\}$. It was shown in [14, 12] that if S is a semilinear set then $\boldsymbol{u} \in S^*$ can be expressed as Presburger arithmetic formula. In particular, formula (5) is equivalent to

$$
\begin{aligned}
\exists \mu_1, \ldots, \mu_q, \nu_{11}, \ldots, \nu_{qq_q} \in \mathbb{N}.\ & \boldsymbol{u}^Z = \textstyle\sum_{i=1}^{q} (\mu_i \boldsymbol{a}_i + \sum_{j=1}^{q_i} \nu_{ij} \boldsymbol{b}_{ij}) \wedge \\
& \bigwedge_{i=1}^{q} (\mu_i = 0 \to \textstyle\sum_{j=1}^{q_i} \nu_{ij} = 0) \ \wedge \ (\sum_{i=1}^{q} \mu_i = K)
\end{aligned} \tag{7}
$$

where vectors $\boldsymbol{a}_i$'s and $\boldsymbol{b}_{ij}$ can be computed effectively from A_Z and b^Z.

We next characterize condition (6). Renaming variables and incorporating the boundedness of $\boldsymbol{x}, \delta$ into the linear inequations, we can write such condition in the form

$$(\boldsymbol{u}^Q, \Delta) \in K\{(\boldsymbol{x}, \delta) \mid A \cdot (\boldsymbol{x}, \delta) \leq \boldsymbol{b}, \delta > 0\} \tag{8}$$

Here $A \cdot (\boldsymbol{x}, \delta) \leq b$ subsumes the conditions $\boldsymbol{0} \leq \boldsymbol{x} \leq \boldsymbol{1}$, $0 \leq \delta \leq 1$. From the theory of linear programming [17] it follows that the set $\{(\boldsymbol{x}, \delta) \mid A \cdot (\boldsymbol{x}, \delta) \leq \boldsymbol{b}\}$ is a polyhedron, and because the solution set is bounded, it is in fact a polytope. Therefore, there exist finitely many vertices $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_n \in \mathbb{Q}_{[0,1]}^{d}$ for some d such that $A \cdot (\boldsymbol{x}, \delta) \leq \boldsymbol{b}$ is equivalent to

$$\exists \lambda_1, \ldots, \lambda_n \in \mathbb{Q}_{[0,1]}.\ \sum_{i=1}^{n} \lambda_i = 1 \ \wedge \ (\boldsymbol{x}, \delta) = \sum_{i=1}^{n} \lambda_i \boldsymbol{c}_i$$

Consequently, (8) is equivalent to

$$\exists \boldsymbol{u}_1,\dots,\boldsymbol{u}_K.\ (\boldsymbol{u}^Q,\Delta) = \sum_{j=1}^{K} \boldsymbol{u}_j \ \wedge\ \exists \lambda_{11},\dots,\lambda_{Kn}.$$
$$\bigwedge_{j=1}^{K} \Big(\bigwedge_{i=1}^{n} \lambda_{ij} \geq 0 \ \wedge\ \sum_{i=1}^{n} \lambda_{ij} = 1 \ \wedge\ (\boldsymbol{u}_j,\delta_j) = \sum_{i=1}^{n} \lambda_{ij} \boldsymbol{c}_i \wedge \delta_j > 0 \Big) \tag{9}$$

It remains to show that the above condition is equivalent to

$$\exists \beta_1,\dots,\beta_n.\ \big((K=0 \wedge \Delta = 0 \wedge \boldsymbol{u} = \boldsymbol{0}) \vee$$
$$(K \geq 1 \wedge \Delta > 0 \wedge (\boldsymbol{u}^Q,\Delta) = \sum_{i=1}^{n} \beta_i \boldsymbol{c}_i \ \wedge\ \bigwedge_{i=1}^{n} \beta_i \geq 0 \ \wedge\ \sum_{i=1}^{n} \beta_i = K)\big) \tag{10}$$

Case $K = 0$ is trivial, so assume $K \neq 0$. Consider a solution of (9). Letting $\beta_i = \sum_{j=1}^{K} \lambda_{ij}$ we obtain a solution of (10). Conversely, consider a solution of (10). Letting $\alpha_{ij} = \beta_i/K$, $\boldsymbol{u}_j = \boldsymbol{u}/K$, $\delta_j = \Delta/n$ we obtain a solution of (9). This shows the equivalence of (9) and (10).

Conjoining formulas (10) and (7) we complete the proof of Theorem 2. ∎

Satisfiability checking for collection formulas. Because star elimination (as well as the preparatory steps in Section 4) introduce only existential quantifiers, and the satisfiability of MLIRA formulas is decidable (see e.g. [4,3]), we obtain the decidability of the initial formula $G(\boldsymbol{r},\boldsymbol{w}) \wedge \boldsymbol{w} \in \{\boldsymbol{x} \mid F(\boldsymbol{x})\}^*$. Thanks to transformation to sum normal form and Theorem 1, we obtain the decidability of formulas involving sets, multisets and fuzzy sets.

6 Language with Nested Star Operators and Quantifiers

$$\begin{aligned} F &::= A \mid F \wedge F \mid \neg F \mid \exists x.F \\ A &::= t \leq t \mid t{=}t \mid (t,\dots,t){\in}\{(t,\dots,t)|F\}^* \\ t &::= k \mid K \mid t+t \mid K \cdot t \mid \lfloor t \rfloor \mid \mathsf{ite}(F,t,t) \end{aligned}$$

Fig. 5. Syntax of Constraints with Nested Stars and Quantifiers

Theorem 2 can be combined with quantifier elimination for MLIRA formulas [18] to decide a language that permits nested uses of quantifiers and stars. Figure 5 summarizes the syntax of one such language. Note that the expression $(r_1,\dots,r_n){\in}\{(t_1,\dots,t_n)|F\}^*$ has the same meaning as before and its only free variables are in $r_1,\dots,r_n$ (the variables in $\{(t_1,\dots,t_n)|F\}$ are all bound). To decide constraints in this language, we eliminate stars and quantifiers starting from the innermost ones. If the innermost operator is a quantifier, we eliminate it as in [18]. If the innermost operator is a star, we use results of Section 4 and Theorem 2 while keeping all existential quantifiers explicitly to preserve equivalence of the subformula. We obtain an existentially quantifier subformula without stars. We eliminate the generated existential quantifiers by again applying quantifier elimination [18]. Repeating this method we obtain a quantifier-free formula without stars, whose satisfiability can be checked [4, 3].

The language of Figure 5 can be further generalized to allow atomic formulas of the form $(t, \ldots, t) \in S$ where the syntax of S is given by

$$S ::= \{(t, \ldots, t) | F\} \mid S \cup S \mid S \setminus S \mid S + S \mid t \cdot S \mid S^*$$

The basic idea is to flatten such set expressions, eliminate operators $\cup$, $\setminus$, $+$ using their definition, and eliminate S^* using the algorithms we just described. The case of $t \cdot S$ is similar to S^* but the value K from Theorem 2 is fixed and given by term t, as opposed to being existentially quantified.

7 Related Work

Logical constraints on collections that do not support cardinality bounds have been studied in the past. Zarba [20] considered decision procedures for quantifier-free multisets but without the cardinality operator, showing that it reduces to quantifier-free pointwise reasoning. The cardinality operator makes that reduction impossible. Notions of the cardinality operator naturally arising from the Feferman-Vaught theorem [6] can express only a finite amount of information for each element $e \in E$, so they are appropriate only for cardinality sets or for the cardinality of the support of the multisets or a fuzzy set. Recently, Lugiez [10] shows the decidability of constraints with a weaker form of such a limited cardinality operator that counts only distinct elements in a multiset, and shows decidability of certain quantifier-free expressible constraints with cardinality operator.

Note that, because our Theorem 1 is only equisatisfiability and not equivalence, we do not obtain decidability of constraints with quantified collections. In fact, although quantified sets with cardinality bounds are decidable [6, 8], quantified multisets with cardinality bounds are undecidable [12, Section 6].

The work in this paper is based on previous results for the special cases of sets [9] and multisets [14, 13, 12]. We rely on the fact that solutions of formulas of Presburger arithmetic are semilinear sets [7]. Bounds on generators of such sets are presented in [15].

Techniques for deciding formulas of MLIRA formulas are part of implementations of modern satisfiability modulo theory theorem provers [4,3,1] and typically use SAT solving techniques along with techniques from mixed integer-linear programming, or the Omega test [16].

8 Conclusions

We have shown decidability of a rich logic for reasoning about collections. The logic is expressive enough for reasoning about sets, multisets, and fuzzy sets as well as their cardinality bounds. Our results also show that star, much like quantifiers, is a natural operator of MLIRA formulas and can also be eliminated. A direct application of our star elimination technique creates an exponentially larger MLIRA formula. We leave for future work the question whether it is possible to generate polynomially large equisatisfiable formulas as for multisets [13].

Acknowledgements. We thank Nikolaj Bjørner for useful discussions.

References

1. Sergey Berezin, Vijay Ganesh, and David L. Dill. An online proof-producing decision procedure for mixed-integer linear arithmetic. In *TACAS*, 2003.
2. Aaron R. Bradley and Zohar Manna. *The Calculus of Computation*. Springer, 2007.
3. Bruno Dutertre and Leonardo de Moura. A Fast Linear-Arithmetic Solver for DPLL(T). In *CAV*, volume 4144 of *LNCS*, 2006.
4. Bruno Dutertre and Leonardo de Moura. Integrating Simplex with DPLL(T). Technical Report SRI-CSL-06-01, SRI International, 2006.
5. Friedrich Eisenbrand and Gennady Shmonin. Carathéodory bounds for integer cones. *Operations Research Letters*, 34(5):564–568, September 2006. http://dx.doi.org/10.1016/j.orl.2005.09.008.
6. S. Feferman and R. L. Vaught. The first order properties of products of algebraic systems. *Fundamenta Mathematicae*, 47:57–103, 1959.
7. S. Ginsburg and E. Spanier. Semigroups, Pressburger formulas and languages. *Pacific Journal of Mathematics*, 16(2):285–296, 1966.
8. Viktor Kuncak, Hai Huu Nguyen, and Martin Rinard. Deciding Boolean Algebra with Presburger Arithmetic. *J. of Automated Reasoning*, 2006. http://dx.doi.org/10.1007/s10817-006-9042-1.
9. Viktor Kuncak and Martin Rinard. Towards efficient satisfiability checking for Boolean Algebra with Presburger Arithmetic. In *CADE-21*, 2007.
10. D. Lugiez. Multitree automata that count. *Theor. Comput. Sci.*, 333(1-2):225–263, 2005.
11. Denis Lugiez and Silvano Dal Zilio. Multitrees Automata, Presburger's Constraints and Tree Logics. Research report 08-2002, LIF, Marseille, France, June 2002. http://www.lif-sud.univ-mrs.fr/Rapports/08-2002.html.
12. Ruzica Piskac and Viktor Kuncak. Decision procedures for multisets with cardinality constraints. In *VMCAI*, number 4905 in LNCS, 2008.
13. Ruzica Piskac and Viktor Kuncak. Linear arithmetic with stars. In *CAV*, 2008.
14. Ruzica Piskac and Viktor Kuncak. On linear arithmetic with stars. Technical Report LARA-REPORT-2008-005, EPFL, 2008.
15. Loïc Pottier. Minimal solutions of linear diophantine systems: Bounds and algorithms. In *RTA*, volume 488 of *LNCS*, 1991.
16. William Pugh. The Omega test: a fast and practical integer programming algorithm for dependence analysis. In *ACM/IEEE conf. Supercomputing*, 1991.
17. Alexander Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, 1998.
18. Volker Weispfenning. Mixed real-integer linear quantifier elimination. In *ISSAC*, pages 129–136, 1999.
19. L. A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.
20. Calogero G. Zarba. Combining multisets with integers. In *CADE-18*, 2002.